**Iterator Pattern Example: Shape**

**Class: Shape**

**package** iterator.shape;

**public** **class** Shape {

**private** **int** id;

**private** String name;

**public** Shape(**int** id, String name){

**this**.id = id;

**this**.name = name;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

@Override

**public** String toString(){

**return** "ID: "+id+" Shape: "+name;

}

}

**Class: ShapeStorage**

**package** iterator.shape;

**public** **class** ShapeStorage {

**private** Shape []shapes = **new** Shape[5];

**private** **int** index;

**public** **void** addShape(String name){

**int** i = index++;

shapes[i] = **new** Shape(i,name);

}

**public** Shape[] getShapes(){

**return** shapes;

}

}

**Class: ShapeIterator**

**package** iterator.shape;

**import** java.util.Iterator;

**public** **class** ShapeIterator **implements** Iterator<Shape>{

**private** Shape [] shapes;

**int** pos;

**public** ShapeIterator(Shape []shapes){

**this**.shapes = shapes;

}

@Override

**public** **boolean** hasNext() {

**if**(pos >= shapes.length || shapes[pos] == **null**)

**return** **false**;

**return** **true**;

}

@Override

**public** Shape next() {

**return** shapes[pos++];

}

@Override

**public** **void** remove() {

**if**(pos <=0 )

**throw** **new** IllegalStateException("Illegal position");

**if**(shapes[pos-1] !=**null**){

**for** (**int** i= pos-1; i<(shapes.length-1);i++){

shapes[i] = shapes[i+1];

}

shapes[shapes.length-1] = **null**;

}

}

}

**Class: TestIteratorPattern**

**package** iterator.shape;

**public** **class** TestIteratorPattern {

**public** **static** **void** main(String[] args) {

ShapeStorage storage = **new** ShapeStorage();

storage.addShape("Polygon");

storage.addShape("Hexagon");

storage.addShape("Circle");

storage.addShape("Rectangle");

storage.addShape("Square");

ShapeIterator iterator = **new** ShapeIterator(storage.getShapes());

**while**(iterator.hasNext()){

System.***out***.println(iterator.next());

}

System.***out***.println("Apply removing while iterating...");

iterator = **new** ShapeIterator(storage.getShapes());

**while**(iterator.hasNext()){

System.***out***.println(iterator.next());

iterator.remove();

}

}

}

Result:

ID: 0 Shape: Polygon

ID: 1 Shape: Hexagon

ID: 2 Shape: Circle

ID: 3 Shape: Rectangle

ID: 4 Shape: Square

Apply removing while iterating...

ID: 0 Shape: Polygon

ID: 2 Shape: Circle

ID: 4 Shape: Square